

```

PPPPPPPPPPPP      AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTT      LLL
PPPPPPPPPPPPPP    AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTT      LLL
PPPPPPPPPPPPPP    AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTT      LLL
PPP                PPP  AAA          AAA  SSS                RRR                RRR                TTT                LLL
PPP                PPP  AAA          AAA  SSS                RRR                RRR                TTT                LLL
PPP                PPP  AAA          AAA  SSS                RRR                RRR                TTT                LLL
PPP                PPP  AAA          AAA  SSS                RRR                RRR                TTT                LLL
PPP                PPP  AAA          AAA  SSS                RRR                RRR                TTT                LLL
PPP                PPP  AAA          AAA  SSS                RRR                RRR                TTT                LLL
PPPPPPPPPPPPPP    AAA          AAA          SSSSSSSSSS      RRRRRRRRRRRR      TTT                LLL
PPPPPPPPPPPPPP    AAA          AAA          SSSSSSSSSS      RRRRRRRRRRRR      TTT                LLL
PPPPPPPPPPPPPP    AAA          AAA          SSSSSSSSSS      RRRRRRRRRRRR      TTT                LLL
PPP                AAAAAAAAAAAAAAAAAA          SSS                RRR                RRR                TTT                LLL
PPP                AAAAAAAAAAAAAAAAAA          SSS                RRR                RRR                TTT                LLL
PPP                AAAAAAAAAAAAAAAAAA          SSS                RRR                RRR                TTT                LLL
PPP                AAA          AAA          SSS                RRR                RRR                TTT                LLL
PPP                AAA          AAA          SSS                RRR                RRR                TTT                LLL
PPP                AAA          AAA          SSS                RRR                RRR                TTT                LLL
PPP                AAA          AAA          SSS                RRR                RRR                TTT                LLL
PPP                AAA          AAA          SSSSSSSSSSSS      RRR                RRR                TTT                LLLLLLLLLLLLLLLLLL
PPP                AAA          AAA          SSSSSSSSSSSS      RRR                RRR                TTT                LLLLLLLLLLLLLLLLLL
PPP                AAA          AAA          SSSSSSSSSSSS      RRR                RRR                TTT                LLLLLLLLLLLLLLLLLL

```

—\$2

Sym

PAS

PAS
PASPAS
PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PAS

PAS
PAS

PAS

PAS

PAS

PAS

PAC

PAS

PAS

PAS
PAS

PAS

PAS

PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PAS

PAS

```
PPPPPPPP      AAAAAA      SSSSSSSS      GGGGGGGG      EEEEEEEEEEE      TTTTTTTTTTT
PPPPPPPP      AAAAAA      SSSSSSSS      GGGGGGGG      EEEEEEEEEEE      TTTTTTTTTTT
PP      PP      AA      AA      SS      GG      EE      TT
PP      PP      AA      AA      SS      GG      EE      TT
PP      PP      AA      AA      SS      GG      EE      TT
PP      PP      AA      AA      SS      GG      EE      TT
PPPPPPPP      AA      AA      SSSSSS      GG      EEEEEEEEE      TT
PPPPPPPP      AA      AA      SSSSSS      GG      EEEEEEEEE      TT
PP      AAAAAAAAAA      SS      GG      GG      EE      TT
PP      AAAAAAAAAA      SS      GG      GG      EE      TT
PP      AA      AA      SS      GG      GG      EE      TT
PP      AA      AA      SSSSSSSS      GG      GG      EE      TT
PP      AA      AA      SSSSSSSS      GGGGGG      EEEEEEEEEEE      TT
PP      AA      AA      SSSSSSSS      GGGGGG      EEEEEEEEEEE      TT
                                     ....
                                     ....
                                     ....

LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
```



```
1 0001 0 MODULE PAS$GET ( %TITLE 'GET procedure'
2 0002 0 IDENT = '1-002' ! File: PASGET.B32 Edit: SBL1002
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY: Pascal Language Support
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module contains PAS$GET, which implements the
36 0036 1 VAX-11 Pascal GET procedure.
37 0037 1
38 0038 1 ENVIRONMENT: User mode - AST reentrant
39 0039 1
40 0040 1 AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. SBL 1-April-1981
45 0045 1 1-002 - Turn off NLK bit after $GET. SPR 11-55690 SBL 8-Apr-1983
46 0046 1 --
47 0047 1
```

PASSGET
1-002

GET procedure
Declarations

I 12
16-Sep-1984 01:37:43
14-Sep-1984 12:51:30

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASGET.B32;1

Page 2
(2)

```

: 49      0048 1 %SBTTL 'Declarations'
: 50      0049 1
: 51      0050 1 PROLOGUE DEFINITIONS:
: 52      0051 1
: 53      0052 1
: 54      0053 1 REQUIRE 'RTLIN:PASPROLOG';           ! Externals, linkages, PSECTs, structures
: 55      0117 1
: 56      0118 1
: 57      0119 1 TABLE OF CONTENTS:
: 58      0120 1
: 59      0121 1
: 60      0122 1 FORWARD ROUTINE
: 61      0123 1     PASSGET: NOVALUE,                ! Do a GET
: 62      0124 1     PASSGET UNLOCK: NOVALUE,          ! Do a GET and UNLOCK
: 63      0125 1     PASS$GET: JSB_GET NOVALUE;        ! Internally callable
: 64      0126 1
: 65      0127 1
: 66      0128 1 MACROS:
: 67      0129 1
: 68      0130 1     NONE
: 69      0131 1
: 70      0132 1 EQUATED SYMBOLS:
: 71      0133 1
: 72      0134 1     NONE
: 73      0135 1
: 74      0136 1 FIELDS:
: 75      0137 1
: 76      0138 1     NONE
: 77      0139 1
: 78      0140 1 OWN STORAGE:
: 79      0141 1
: 80      0142 1     NONE
```


PASSGET
1-002

GET procedure
PASSGET - GET procedure

J 12
16-Sep-1984 01:37:43
14-Sep-1984 12:51:30

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASGET.B32;1

Page 3
(3)

```

82 0143 1 %SBTTL 'PASSGET - GET procedure'
83 0144 1 GLOBAL ROUTINE PASSGET (
84 0145 1 PFV: REF $PASSPFV_FILE_VARIABLE,
85 0146 1 ERROR
86 0147 1 ): NOVALUE =
87 0148 1
88 0149 1 ++
89 0150 1 FUNCTIONAL DESCRIPTION:
90 0151 1
91 0152 1 PASSGET implements the VAX-11 Pascal GET procedure. It
92 0153 1 reads a record from the file and places its contents in
93 0154 1 the file buffer.
94 0155 1
95 0156 1 CALLING SEQUENCE:
96 0157 1
97 0158 1 CALL PASSGET (PFV.mr.r [, ERROR.j.r])
98 0159 1
99 0160 1 FORMAL PARAMETERS:
100 0161 1
101 0162 1 PFV - The Pascal File Variable (PFV) passed by reference.
102 0163 1 The structure of the PFV is defined in PASPFV.REQ.
103 0164 1
104 0165 1 ERROR - Optional. If specified, the address to unwind to
105 0166 1 if an error occurs.
106 0167 1
107 0168 1 IMPLICIT INPUTS:
108 0169 1
109 0170 1 NONE
110 0171 1
111 0172 1 IMPLICIT OUTPUTS:
112 0173 1
113 0174 1 NONE
114 0175 1
115 0176 1 ROUTINE VALUE:
116 0177 1
117 0178 1 NONE
118 0179 1
119 0180 1 SIDE EFFECTS:
120 0181 1
121 0182 1 Opens standard files INPUT or OUTPUT if not already open.
122 0183 1 Turns off RAB$V_NLK which might have been set by PASSGET_UNLOCK
123 0184 1
124 0185 1 SIGNALLED ERRORS:
125 0186 1
126 0187 1 FILNOTOPE - file not open
127 0188 1 FILNOTINS - file not in Inspection mode
128 0189 1
129 0190 1 --
130 0191 1
131 0192 2 BEGIN
132 0193 2
133 0194 2 LOCAL
134 0195 2 FCB: REF $PASSFCB_CONTROL_BLOCK,
135 0196 2 PFV_ADDR: VOLATILE,
136 0197 2 UNWIND_ACT: VOLATILE,
137 0198 2 ERROR_ADDR: VOLATILE;
138 0199 2
```

! Do a GET
! File variable
! Unwind address if error

! File control block
! Enable argument
! Enable argument
! Enable argument

```
139 0200 2 BIND
140 0201 2 RAB = FCB: REF BLOCK [, BYTE]; ! RAB is FCB
141 0202 2
142 0203 2 BUILTIN
143 0204 2 ACTUALCOUNT,
144 0205 2 TESTBITSS;
145 0206 2
146 0207 2 ENABLE
147 0208 2 PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);
148 0209 2
149 0210 2 IF ACTUALCOUNT () GEQU 2
150 0211 2 THEN
151 0212 2 ERROR_ADDR = .ERROR; ! Set unwind address
152 0213 2
153 0214 2 !+
154 0215 2 ! Set PFV address enable argument.
155 0216 2 !-
156 0217 2
157 0218 2 PFV_ADDR = PFV [PFV$R_PFV];
158 0219 2
159 0220 2 !+
160 0221 2 ! Validate and lock PFV
161 0222 2 !-
162 0223 2
163 0224 2 PASS$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
164 0225 2
165 0226 2 !+
166 0227 2 ! Set unwind action to unlock file.
167 0228 2 !-
168 0229 2
169 0230 2 UNWIND_ACT = PASS$K_UNWIND_UNLOCK;
170 0231 2
171 0232 2 !+
172 0233 2 ! Resolve lazy lookahead, if any. Also may open INPUT or OUTPUT.
173 0234 2 !-
174 0235 2
175 0236 2 IF NOT .PFV [PFV$V_VALID]
176 0237 2 THEN
177 0238 2 PASS$LOOK_AHEAD (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
178 0239 2
179 0240 2 !+
180 0241 2 ! Verify that the file is open.
181 0242 2 !-
182 0243 2
183 0244 2 IF NOT .PFV [PFV$V_OPEN]
184 0245 2 THEN
185 0246 2 $PASS$IO_ERROR (PASS$_FILNOTOPE,0);
186 0247 2
187 0248 2 !+
188 0249 2 ! Verify that we are in Inspection mode
189 0250 2 !-
190 0251 2
191 0252 2 IF NOT .FCB [FCB$V_INSPECTION]
192 0253 2 THEN
193 0254 2 $PASS$IO_ERROR (PASS$_FILNOTINS,0);
194 0255 2
195 0256 2 !+
```


PASSGET
1-002

GET procedure
PASSGET - GET procedure

L 12
16-Sep-1984 01:37:43
14-Sep-1984 12:51:30

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASGET.B32;1

Page 5
(3)

```
: 196      0257 2      ! Turn off RAB$V_NLK.
: 197      0258 2      !-
: 198      0259 2      RAB [RAB$V_NLK] = 0;
: 199      0260 2
: 200      0261 2
: 201      0262 2      !+
: 202      0263 2      ! Do the GET
: 203      0264 2      !-
: 204      0265 2
: 205      0266 2      PASS$GET (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);
: 206      0267 2
: 207      0268 2      !+
: 208      0269 2      ! Indicate successful completion
: 209      0270 2      ! Unlock the file
: 210      0271 2      !-
: 211      0272 2
: 212      0273 2      FCB [FCB$L_STATUS] = 0;
: 213      0274 2      PFV [PFV$V_LOCK] = 0;
: 214      0275 2
: 215      0276 2      RETURN;
: 216      0277 2
: 217      0278 1      END;
```

! End of routine PASSGET

```
.TITLE PASSGET GET procedure
.IDENT \1-002\

.EXTRN PASSGET, PASSGET_UNLOCK
.EXTRN PASS$GET, PASS$IO_HANDLER
.EXTRN PASS$VALIDATE_PFV
.EXTRN PASS$LOOK_AHEAD
.EXTRN PASS$SIGNAL, PASSK_FILNOTOPE
.EXTRN PASSK_FILNOTINS
```

.PSECT _PASSCODE, NOWRT, SHR, PIC, 2

```
.ENTRY PASSGET, Save R2,R3,R6,R7
SUBL2 #8, SP
CLRL ERROR_ADDR
CLRQ UNWIND_ACT
MOVAL 6$, (FP)
CMPB (AP), #2
BLSSU 1$
MOVL ERROR, ERROR_ADDR
MOVL PFV, R6
MOVL R6, PFV_ADDR
JSB PASS$VALIDATE_PFV
MOVL #1, UNWIND_ACT
BLBS 6(R6), 2$
JSB PASS$LOOK_AHEAD
BBS #5, 7(R6), 3$
CLRL -(SP)
MOVZBL #PASSK_FILNOTOPE, -(SP)
BRB 4$
BBS #3, -3(FCB), 5$
CLRL -(SP)
MOVZBL #PASSK_FILNOTINS, -(SP)
```

```
00CC 00000
5E      08 C2 00002
      7E D4 00005
      04 AE 7C 00007
6D      04 CF DE 0000A
02      6C 91 0000F
      04 1F 00012
6E      08 AC D0 00014
56      04 AC D0 00018 1$:
08 AE 56 D0 0001C
      04 AE 00000000G 00 16 00020
      06 AE 01 D0 00026
      06 A6 E8 0002A
08 07 A6 00000000G 00 16 0002E
      05 E0 00034 2$:
      7E 00G 7E D4 00039
      08 8F 9A 0003B
      0E FD A7 0B 11 0003F
      7E 00G 03 E0 00041 3$:
      7E 00G 7E D4 00046
      7E 00G 8F 9A 00048
```

```
: 0144
: 0192
: 0210
: 0212
: 0218
: 0224
: 0230
: 0236
: 0238
: 0244
: 0246
: 0252
: 0254
```

PASSGET
1-002

GET procedure
PASSGET - GET procedure

M 12
16-Sep-1984 01:37:43 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:51:30 [PASRTL.SRC]PASGET.B32;1

Page 6
(3)

00000000G	00	02	FB	0004C	4\$:	CALLS	#2, PASS\$SIGNAL	:
			04	00053		RET		:
06	A7	10	8A	00054	5\$:	BICB2	#16, 6(FCB)	: 0260
		0000V	30	00058		BSBW	PASS\$GET	: 0266
		D4	A7	D4	0005B	CLRL	-44(FCB)	: 0273
07	A6	80	8F	8A	0005E	BICB2	#128, 7(R6)	: 0274
				04	00063	RET		: 0278
				0000	00064	6\$:	.WORD	: 0192
	50	08	AC	D0	00066	MOVL	8(AP), R0	:
	50	04	A0	D0	0006A	MOVL	4(R0), R0	:
		F4	A0	9F	0006E	PUSHAB	ERROR_ADDR	:
		F8	A0	9F	00071	PUSHAB	UNWIND_ACT	:
		FC	A0	9F	00074	PUSHAB	PFV_ADDR	:
			03	DD	00077	PUSHL	#3	:
			5E	DD	00079	PUSHL	SP	:
	7E	04	AC	7D	0007B	MOVQ	4(AP), -(SP)	:
00000000G	00		03	FB	0007F	CALLS	#3, PASS\$IO_HANDLER	:
			04	00086		RET		:

; Routine Size: 135 bytes, Routine Base: _PASS\$CODE + 0000

; 218 0279 1
; 219 0280 1 !<BLF/PAGE>

PASSGET
1-002

GET procedure
PASSGET_UNLOCK - GET procedure with UNLOCK

N 12
16-Sep-1984 01:37:43
14-Sep-1984 12:51:30

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASGET.B32;1

Page 7
(4)

```

221 0281 1 %SBTTL 'PASSGET_UNLOCK - GET procedure with UNLOCK'
222 0282 1 GLOBAL ROUTINE PASSGET_UNLOCK (
223 0283 1     PFV: REF $PASSPFV_FILE_VARIABLE,           ! File variable
224 0284 1     ERROR                                     ! Unwind address if error
225 0285 1 ): NOVALUE =
226 0286 1
227 0287 1 ++
228 0288 1 FUNCTIONAL DESCRIPTION:
229 0289 1
230 0290 1     PASSGET_UNLOCK is called by the compiled code when a READ is
231 0291 1     done for a non-textfile. It does the equivalent of a GET followed
232 0292 1     by an UNLOCK. What it actually does is set RAB$V_NLK which will
233 0293 1     cause the next $GET to not lock the record. This bit is reset by
234 0294 1     PASS$FIND2, PASS$FINDK, PASS$GET, PASS$RESET and PASS$RESETK and
235 0295 1     PASS$REWRITE2, which are the only procedures
236 0296 1     which do $GETs on non-textfiles. It doesn't matter for textfiles
237 0297 1     since they may only be sequential organization.
238 0298 1
239 0299 1 CALLING SEQUENCE:
240 0300 1
241 0301 1     CALL PASSGET_UNLOCK (PFV.mr.r [, ERROR.j.r])
242 0302 1
243 0303 1 FORMAL PARAMETERS:
244 0304 1
245 0305 1     PFV           - The Pascal File Variable (PFV) passed by reference.
246 0306 1                   The structure of the PFV is defined in PASPFV.REQ.
247 0307 1
248 0308 1     ERROR        - Optional. If specified, the address to unwind to
249 0309 1                   if an error occurs.
250 0310 1
251 0311 1 IMPLICIT INPUTS:
252 0312 1
253 0313 1     NONE
254 0314 1
255 0315 1 IMPLICIT OUTPUTS:
256 0316 1
257 0317 1     NONE
258 0318 1
259 0319 1 ROUTINE VALUE:
260 0320 1
261 0321 1     NONE
262 0322 1
263 0323 1 SIDE EFFECTS:
264 0324 1
265 0325 1     Opens standard files INPUT or OUTPUT if not already open.
266 0326 1
267 0327 1 SIGNALLED ERRORS:
268 0328 1
269 0329 1     FILNOTOPE - file not open
270 0330 1     FILNOTINS - file not in Inspection mode
271 0331 1
272 0332 1 --
273 0333 1
274 0334 2 BEGIN
275 0335 2
276 0336 2 LOCAL
277 0337 2     FCB: REF $PASSFCB_CONTROL_BLOCK,           ! File control block
```

```
: 278      0338      2      PFV_ADDR: VOLATILE,      ! Enable argument
: 279      0339      2      UNWIND_ACT: VOLATILE,      ! Enable argument
: 280      0340      2      ERROR_ADDR: VOLATILE;      ! Enable argument
: 281      0341      2
: 282      0342      2      BIND
: 283      0343      2      RAB = FCB: REF BLOCK [, BYTE];      ! RAB is FCB address
: 284      0344      2
: 285      0345      2      BUILTIN
: 286      0346      2      ACTUALCOUNT,
: 287      0347      2      TESTBITSS;
: 288      0348      2
: 289      0349      2      ENABLE
: 290      0350      2      PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);
: 291      0351      2
: 292      0352      2      IF ACTUALCOUNT () GEQU 2
: 293      0353      2      THEN
: 294      0354      2      ERROR_ADDR = .ERROR;      ! Set unwind address
: 295      0355      2
: 296      0356      2      !+
: 297      0357      2      ! Set PFV address enable argument.
: 298      0358      2      !-
: 299      0359      2
: 300      0360      2      PFV_ADDR = PFV [PFV$R_PFV];
: 301      0361      2
: 302      0362      2      !+
: 303      0363      2      ! Validate and lock PFV
: 304      0364      2      !-
: 305      0365      2
: 306      0366      2      PASS$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
: 307      0367      2
: 308      0368      2      !+
: 309      0369      2      ! Set unwind action to unlock file.
: 310      0370      2      !-
: 311      0371      2
: 312      0372      2      UNWIND_ACT = PASS$K_UNWIND_UNLOCK;
: 313      0373      2
: 314      0374      2      !+
: 315      0375      2      ! Resolve lazy lookahead, if any. Also may open INPUT or OUTPUT.
: 316      0376      2      !-
: 317      0377      2
: 318      0378      2      IF NOT .PFV [PFV$V_VALID]
: 319      0379      2      THEN
: 320      0380      2      PASS$LOOK_AHEAD (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
: 321      0381      2
: 322      0382      2      !+
: 323      0383      2      ! Verify that the file is open.
: 324      0384      2      !-
: 325      0385      2
: 326      0386      2      IF NOT .PFV [PFV$V_OPEN]
: 327      0387      2      THEN
: 328      0388      2      $PASS$IO_ERROR (PASS$_FILNOTOPE,0);
: 329      0389      2
: 330      0390      2      !+
: 331      0391      2      ! Verify that we are in Inspection mode
: 332      0392      2      !-
: 333      0393      2
: 334      0394      2      IF NOT .FCB [FCB$V_INSPECTION]
```


PASSGET
1-002

GET procedure
PASSGET_UNLOCK - GET procedure with UNLOCK

C 13
16-Sep-1984 01:37:43
14-Sep-1984 12:51:30

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASGET.B32;1

Page 9
(4)

```

: 335 0395 2 THEN
: 336 0396 2 $PASSIO_ERROR (PASS$_FILNOTINS,0);
: 337 0397 2
: 338 0398 2
: 339 0399 2 !+
: 340 0400 2 ! If this is not a textfile, set RAB$_NLK.
: 341 0401 2 !-
: 342 0402 2 IF NOT .FCB [FCB$_TEXT]
: 343 0403 2 THEN
: 344 0404 2 RAB [RAB$_NLK] = 1;
: 345 0405 2
: 346 0406 2 !+
: 347 0407 2 ! Do the GET
: 348 0408 2 !-
: 349 0409 2
: 350 0410 2 PASS$GET (PFV [PFV$_PFV], FCB [FCB$_FCB]);
: 351 0411 2
: 352 0412 2 !+
: 353 0413 2 ! Indicate successful completion
: 354 0414 2 ! Unlock the file
: 355 0415 2 !-
: 356 0416 2
: 357 0417 2 FCB [FCB$_STATUS] = 0;
: 358 0418 2 PFV [PFV$_LOCK] = 0;
: 359 0419 2
: 360 0420 2 RETURN;
: 361 0421 2
: 362 0422 1 END;
```

! End of routine PASSGET_UNLOCK

			00CC	00000	.ENTRY	PASSGET_UNLOCK, Save R2,R3,R6,R7	: 0282
	5E		08	C2	SUBL2	#8, SP	: 0334
			7E	D4	CLRL	ERROR_ADDR	: 0352
		04	AE	7C	CLRL	UNWIND_ACT	: 0354
	6D	005A	CF	DE	MOVAL	7\$, (FP)	: 0360
	02		6C	91	CMPB	(AP), #2	: 0366
			04	1F	BLSSU	1\$	: 0372
	6E	08	AC	D0	MOVL	ERROR, ERROR_ADDR	: 0378
	56	04	AC	D0	MOVL	PFV, R6	: 0380
08	AE		56	D0	MOVL	R6, PFV_ADDR	: 0386
		00000000G	00	16	JSB	PASS\$VACIDATE PFV	: 0388
	04	AE	01	D0	MOVL	#1, UNWIND_ACT	: 0394
	06	06	A6	E8	BLBS	6(R6), 2\$	: 0396
		00000000G	00	16	JSB	PASS\$LOOK_AHEAD	: 0402
08	07	A6	05	E0	BBS	#5, 7(R6), 3\$	: 0402
			7E	D4	CLRL	-(SP)	: 0402
	7E	00G	8F	9A	MOVZBL	#PASS\$_FILNOTOPE, -(SP)	: 0402
			0B	11	BRB	4\$	: 0402
0E	FD	A7	03	E0	BBS	#3, -3(FCB), 5\$	: 0402
			7E	D4	CLRL	-(SP)	: 0402
	7E	00G	8F	9A	MOVZBL	#PASS\$_FILNOTINS, -(SP)	: 0402
	00000000G	00	02	FB	CALLS	#2, PASS\$SIGNAL	: 0402
				04	RET		: 0402
	04	F8	A7	E8	BLBS	-8(FCB), 6\$	: 0402

PASSGET
1-002

GET procedure
PASSGET_UNLOCK - GET procedure with UNLOCK

D 13
16-Sep-1984 01:37:43
14-Sep-1984 12:51:30

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASGET.B32;1

Page 10
(4)

06	A7	1C	88	00058	BISB2	#16, 6(FCB)	: 0404
		0000V	70	0005C	BSBW	PASSGET	: 0410
		D4	A7	D4	CLRL	-44(FCB)	: 0417
07	A6	80	8F	3A	BICB2	#128, 7(R6)	: 0418
			04	00067	RET		: 0422
			0000	00068	.WORD	Save nothing	: 0334
	50	08	AC	D0	MOVL	8(AP), R0	
	50	04	A0	D0	MOVL	4(R0), R0	
		F4	A0	9F	PUSHAB	ERROR_ADDR	
		F8	A0	9F	PUSHAB	UNWIND_ACT	
		FC	A0	9F	PUSHAB	PFV_ADDR	
			03	DD	PUSHL	#3	
			5E	DD	PUSHL	SP	
	7E	04	AC	7D	MOVQ	4(AP), -(SP)	
00000000G	00		03	FB	CALLS	#3, PASSIO_HANDLER	
			04	0008A	RET		

; Routine Size: 139 bytes, Routine Base: _PASSCODE + 0087

: 363 0423 1
: 364 0424 1 !<BLF/PAGE>


```

: 366 0425 1 %SBTTL 'PAS$$GET - Internally callable GET'
: 367 0426 1 GLOBAL ROUTINE PAS$$GET (
: 368 0427 1     PFV: REF $PAS$PFV_FILE_VARIABLE,
: 369 0428 1     FCB: REF $PAS$FCB_CONTROL_BLOCK
: 370 0429 1 ): JSB_GET NOVALUE =
: 371 0430 1
: 372 0431 1 ++
: 373 0432 1 FUNCTIONAL DESCRIPTION:
: 374 0433 1
: 375 0434 1     PAS$$GET implements the VAX-11 Pascal GET procedure. It
: 376 0435 1     reads a record from the file and places its contents in
: 377 0436 1     the file buffer.
: 378 0437 1
: 379 0438 1 CALLING SEQUENCE:
: 380 0439 1
: 381 0440 1     JSB PAS$$GET (PFV.mr.r, FCB.mr.r)
: 382 0441 1
: 383 0442 1 FORMAL PARAMETERS:
: 384 0443 1
: 385 0444 1     PFV           - The Pascal File Variable (PFV) passed by reference.
: 386 0445 1                   The structure of the PFV is defined in PASPFV.REQ.
: 387 0446 1
: 388 0447 1     FCB           - The File Control Block for the file.
: 389 0448 1
: 390 0449 1 IMPLICIT INPUTS:
: 391 0450 1
: 392 0451 1     It is assumed that the file is open, is in inspection mode and is locked.
: 393 0452 1     It is assumed that lazy lookahead is not in progress.
: 394 0453 1
: 395 0454 1 IMPLICIT OUTPUTS:
: 396 0455 1
: 397 0456 1     NONE
: 398 0457 1
: 399 0458 1 ROUTINE VALUE:
: 400 0459 1
: 401 0460 1     NONE
: 402 0461 1
: 403 0462 1 SIDE EFFECTS:
: 404 0463 1
: 405 0464 1     NONE
: 406 0465 1
: 407 0466 1 SIGNALLED ERRORS:
: 408 0467 1
: 409 0468 1     FAIGETLOC - failed to GET locked component
: 410 0469 1     GETAFTEOF - GET attempted after end-of-file
: 411 0470 1     ERRDURGET - error during GET
: 412 0471 1
: 413 0472 1 --
: 414 0473 1
: 415 0474 2 BEGIN
: 416 0475 2
: 417 0476 2 LOCAL
: 418 0477 2     STATUS;
: 419 0478 2
: 420 0479 2 BIND
: 421 0480 2     RAB = FCB: REF BLOCK [, BYTE];
: 422 0481 2
```

```

: 423      0482 2      !+
: 424      0483 2      ! Set initial conditions of PFV flags
: 425      0484 2      !-
: 426      0485 2
: 427      0486 2      PFV [PFV$V_DFB] = 0;      ! Buffer undefined
: 428      0487 2      PFV [PFV$V_EOF_DEFINED] = 1; ! EOF defined after GET
: 429      0488 2
: 430      0489 2
: 431      0490 2      !+
: 432      0491 2      ! If EOF is true, then signal an error.
: 433      0492 2      !-
: 434      0493 2      IF .FCB [FCB$V_EOF]
: 435      0494 2      THEN
: 436      0495 2          $PASSIO_ERROR (PASS_GETAFTEOF,0);
: 437      0496 2
: 438      0497 2      !+
: 439      0498 2      ! GET from a textfile is different...
: 440      0499 2      !-
: 441      0500 2
: 442      0501 2      IF .FCB [FCB$V_TEXT]
: 443      0502 2      THEN
: 444      0503 3          BEGIN
: 445      0504 3              !+
: 446      0505 3              ! Are we already at end-of-line? If so, set lazy-lookahead
: 447      0506 3              !-
: 448      0507 3              IF .PFV [PFV$V_EOLN]
: 449      0508 3              THEN
: 450      0509 4                  BEGIN
: 451      0510 4                      FCB [FCB$V_LAZY] = 1;
: 452      0511 4                      PFV [PFV$V_VALID] = 0;
: 453      0512 4                      RETURN;
: 454      0513 3                  END;
: 455      0514 3
: 456      0515 3              !+
: 457      0516 3              ! Advance current record pointer
: 458      0517 3              !-
: 459      0518 3
: 460      0519 3          FCB [FCB$A_RECORD_CUR] = .FCB [FCB$A_RECORD_CUR] + 1;
: 461      0520 3
: 462      0521 3              !+
: 463      0522 3              ! Are we now at end-of-line?
: 464      0523 3              !-
: 465      0524 3
: 466      0525 3          IF .FCB [FCB$A_RECORD_CUR] GEQA .FCB [FCB$A_RECORD_END]
: 467      0526 3          THEN
: 468      0527 4              BEGIN
: 469      0528 4                  !+
: 470      0529 4                  ! At end-of-line. Set EOL, move a blank to user buffer.
: 471      0530 4                  !-
: 472      0531 4
: 473      0532 4                  PFV [PFV$V_EOLN] = 1;
: 474      0533 4                  CH$WCHAR (%C' ', .PFV [PFV$A_BUFFER]);
: 475      0534 4                  END
: 476      0535 3          ELSE
: 477      0536 3              !+
: 478      0537 3              ! Otherwise move new character to user buffer
: 479      0538 3              !-

```



```

480 0539 3
481 0540
482 0541      CH$WCHAR (..FCB [FCB$A_RECORD_CUR], .PFV [PFV$A_BUFFER]);
483 0542
484 0543      !+
485 0544      ! Mark buffer and PFV as valid and return immediately.
486 0545      !-
487 0546      PFV [PFV$V_VALID] = 1;
488 0547      PFV [PFV$V_DFB] = 1;
489 0548      RETURN;
490 0549      END;
491 0550
492 0551      !+
493 0552      ! Not textfile, call RMS to get a record.
494 0553      !-
495 0554
496 0555      RAB [RAB$B_RAC] = RAB$C_SEQ;      ! Set sequential access mode
497 0556      STATUS = $PAS$RMS_OP ($GET (RAB=.RAB));
498 0557
499 0558      !+
500 0559      ! If varying, put component length in first word of user buffer.
501 0560      ! Do this before we check for errors!
502 0561      !-
503 0562
504 0563      IF .FCB [FCB$V_VARYING]
505 0564      THEN
506 0565          (.PFV [PFV$A_BUFFER])<0,16,0> = .RAB [RAB$W_RSZ];
507 0566
508 0567      !+
509 0568      ! Turn off RAB$V_NLK now just to make sure we don't have problems
510 0569      ! later on.
511 0570      !-
512 0571
513 0572      RAB [RAB$V_NLK] = 0;
514 0573
515 0574      IF NOT .STATUS
516 0575      THEN
517 0576          BEGIN
518 0577              !+
519 0578              ! $GET failed. If RMS$_EOF, simply set EOF condition. Otherwise,
520 0579              ! signal an error.
521 0580              !-
522 0581              IF .RAB [RAB$L_STS] EQLU RMS$_EOF
523 0582              THEN
524 0583                  FCB [FCB$V_EOF] = 1
525 0584              ELSE IF .RAB [RAB$L_STS] EQLU RMS$_RLK
526 0585              THEN
527 0586                  $PAS$IO_ERROR (PAS$_FAIGETLOC)      ! Failed to GET locked component
528 0587              ELSE
529 0588                  $PAS$IO_ERROR (PAS$_ERRDURGET);
530 0589              PFV [PFV$V_DFB] = 0;
531 0590              END
532 0591          ELSE
533 0592              BEGIN
534 0593                  FCB [FCB$V_EOF] = 0;
535 0594                  PFV [PFV$V_DFB] = 1;
536 0595              END;
```

: 537
: 538
: 539
: 540

```
0596 2
0597 2 RETURN;
0598 2
0599 1 END;
```

```
! End of routine PASS$GET
```

```
.EXTRN PASSK_GETAFTEOF
.EXTRN SYSSGET, PASSK_FAIGETLOC
.EXTRN PASSK_ERRDURGET
```

Address	Instruction	Comment	Address	Instruction	Comment
00000000G	02 A2	04 A6 9E 00000 PASS\$GET::	00000000G	02 A2	MOVAB 4(PFV), R2
00000000G	02 A2	04 8A 00004	00000000G	02 A2	BICB2 #2, 2(R2)
00000000G	53 63	04 88 00008	00000000G	53 63	BISB2 #4, 2(R2)
00000000G	7E 00	FC A7 9E 0000C	00000000G	7E 00	MOVAB -4(FCB), R3
00000000G	2C 62	0D E1 00010	00000000G	2C 62	BBC #13, (R3), 1\$
00000000G	01 A3	7E D4 00014	00000000G	01 A3	CLRL -(SP)
00000000G	02 A2	00G 8F 9A 00016	00000000G	02 A2	MOVZBL #PASSK_GETAFTEOF, -(SP)
00000000G	F0 A7	02 FB 0001A	00000000G	F0 A7	CALLS #2, PASS\$SIGNAL
00000000G	02 B6	05 00021	00000000G	02 B6	RSB
00000000G	00 B6	F8 A7 E9 00022 1\$:	00000000G	00 B6	BLBC -8(FCB), 5\$
00000000G	02 A2	13 E1 00026	00000000G	02 A2	BBC #19, (R2), 2\$
00000000G	00 B6	04 88 0002A	00000000G	00 B6	BISB2 #4, 1(R3)
00000000G	02 A2	01 8A 0002E	00000000G	02 A2	BICB2 #1, 2(R2)
00000000G	00 B6	05 00032	00000000G	00 B6	RSB
00000000G	02 A2	EC A7 D6 00033 2\$:	00000000G	02 A2	INCL -20(FCB)
00000000G	00 B6	EC A7 D1 00036	00000000G	00 B6	CMPL -20(FCB), -16(FCB)
00000000G	02 A2	0A 1F 0003B	00000000G	02 A2	BLSSU 3\$
00000000G	00 B6	08 88 0003D	00000000G	00 B6	BISB2 #8, 2(R2)
00000000G	02 A2	20 90 00041	00000000G	02 A2	MOVB #32, a0(PFV)
00000000G	00 B6	05 11 00045	00000000G	00 B6	BRB 4\$
00000000G	02 A2	EC B7 90 00047 3\$:	00000000G	02 A2	MOVB a-20(FCB), a0(PFV)
00000000G	00 B6	01 88 0004C 4\$:	00000000G	00 B6	BISB2 #1, 2(R2)
00000000G	02 A2	62 11 00050	00000000G	02 A2	BRB 14\$
00000000G	00 B6	1E A7 94 00052 5\$:	00000000G	00 B6	CLRB 30(FCB)
00000000G	02 A2	57 DD 00055 6\$:	00000000G	02 A2	PUSHL FCB
00000000G	00 B6	01 FB 00057	00000000G	00 B6	CALLS #1, SYSS\$GET
00000000G	02 A2	50 E8 0005E	00000000G	02 A2	BLBS \$\$\$STATUS, 7\$
00000000G	00 B6	50 D1 00061	00000000G	00 B6	CMPL \$\$\$STATUS, #98906
00000000G	02 A2	04 12 00068	00000000G	02 A2	BNEQ 7\$
00000000G	00 B6	03 A3 E8 0006A	00000000G	00 B6	BLBS 3(R3), 6\$
00000000G	02 A2	02 E1 0006E 7\$:	00000000G	02 A2	BBC #2, -8(FCB), 8\$
00000000G	00 B6	22 A7 B0 00073	00000000G	00 B6	MOVW 34(FCB), a0(PFV)
00000000G	02 A2	10 8A 00078 8\$:	00000000G	02 A2	BICB2 #16, 6(FCB)
00000000G	00 B6	50 E8 0007C	00000000G	00 B6	BLBS STATUS, 13\$
00000000G	02 A2	08 A7 D1 0007F	00000000G	02 A2	CMPL 8(FCB), #98938
00000000G	00 B6	06 12 00087	00000000G	00 B6	BNEQ 9\$
00000000G	02 A2	20 88 00089	00000000G	02 A2	BISB2 #32, 1(R3)
00000000G	00 B6	1C 11 0008D	00000000G	00 B6	BRB 12\$
00000000G	02 A2	08 A7 D1 0008F 9\$:	00000000G	02 A2	CMPL 8(FCB), #98986
00000000G	00 B6	06 12 00097	00000000G	00 B6	BNEQ 10\$
00000000G	02 A2	00G 8F 9A 00099	00000000G	02 A2	MOVZBL #PASSK_FAIGETLOC, -(SP)
00000000G	00 B6	04 11 0009D	00000000G	00 B6	BRB 11\$
00000000G	02 A2	00G 8F 9A 0009F 10\$:	00000000G	02 A2	MOVZBL #PASSK_ERRDURGET, -(SP)
00000000G	00 B6	01 FB 000A3 11\$:	00000000G	00 B6	CALLS #1, PASS\$SIGNAL
00000000G	02 A2	05 000AA	00000000G		

PASSGET
1-002

GET procedure
PASSGET - Internally callable GET

1 13
16-Sep-1984 01:37:43
14-Sep-1984 12:51:30

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASGET.B32;1

Page 15
(5)

02	A2	02	8A 000AB	12\$:	BICB2	#2, 2(R2)
			05 000AF		RSB	
01	A3	20	8A 000B0	13\$:	BICB2	#32, 1(R3)
02	A2	02	88 000B4	14\$:	BISB2	#2, 2(R2)
			05 000B8		RSB	

: 0589
: 0574
: 0593
: 0594
: 0599

; Routine Size: 185 bytes, Routine Base: _PASSCODE + 0112

: 541 0600 1
: 542 0601 1 !<BLF/PAGE>

PASSGET
1-002

GET procedure
PASSGET - Internally callable GET

J 13
16-Sep-1984 01:37:43
14-Sep-1984 12:51:30

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASGET.B32;1

Page 16
(6)

: 544 0602 1 END
: 545 0603 1
: 546 0604 0 ELUDOM

! End of module PASSGET

PSECT SUMMARY

:
: Name Bytes Attributes
: _PASSCODE 459 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

:
: File Total Symbols Loaded Percent Pages Mapped Processing Time
: \$255\$DUA28:[SYSLIB]STARLET.L32;1 9776 12 0 581 00:01.0
: \$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1 427 98 22 33 00:00.4

COMMAND QUALIFIERS

:
: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:PASGET/OBJ=OBJ\$:PASGET MSRC\$:PASGET/UPDATE=(ENH\$:PASGET)

: Size: 459 code + 0 data bytes
: Run Time: 00:11.4
: Elapsed Time: 00:39.4
: Lines/CPU Min: 3181
: Lexemes/CPU-Min: 19359
: Memory Used: 109 pages
: Compilation Complete

0294 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY